

1 **REMARKS**

2 Claims 1, 13, 14, 37, 71 and 94 are amended. Claims 62-63 were  
3 previously cancelled. Claims 1-61 and 64-96 remain in the application for  
4 consideration. In view of the following remarks, Applicant respectfully requests  
5 reconsideration and allowance of the subject application.

6  
7 **35 U.S.C. § 112 Rejections**

8 Claim 94 stands rejected under 35 U.S.C. § 112, second paragraph as being  
9 indefinite. The Office states that the term “sorting the files according to average  
10 order in which the files were downloaded” is confusing as it implies that the files  
11 have already been downloaded.

12 Applicant has amended claim 94 to now recite that the “sorting by file  
13 usage order comprises sorting the files according to the average order in which the  
14 files were downloaded within *scenarios of* their particular priority or priorities.”  
15 As noted in the Specification, a scenario is a script of tasks that the average user  
16 typically follows when using a product during a particular portion of product use.  
17 See, e.g. Specification, page 28, lines 22-24. This, in combination with the  
18 context of the claim (i.e. “A method for ordering files for download to a client...)  
19 should make it apparent that this claim is directed to ordering the files before  
20 download.

21 In view of the amendment and discussion above, Applicant respectfully  
22 submits that there is nothing indefinite with respect to claim 94.  
23  
24  
25

**35 U.S.C. §§ 102 and 103 Rejections**

Claim 82 stands rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,999,740 to Rowley.

Claims 1-5, 8-20, 37-39, 42-46, 55, 57, 61, 62, and 71-76 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of U.S. Patent No. 6,219,698 to Iannucci and a publications entitled “Delphi 5 Developer’s Guide” by Xavier Pacheco et al. (hereinafter “Pacheco”).

Claims 6, 40, 41 and 54 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Iannucci, Pacheco and U.S. Patent No. 4,641,274 to Swank.

Claim 7 stands rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Iannucci, Pacheco and U.S. Patent No. 5,195,183 to Miller.

Claims 21-24, 27, 28, 32, 36, 47, 49-51, 53, and 54 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view Swank.

Claims 25 and 29-31 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Swank and U.S. Patent No. 5,845,090 to Collins III.

Claim 26 stands rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Swank and U.S. Patent No. 6,615,276 to Mastrianni.

Claims 33-35 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Swank and U.S. Patent No. 5,835,777 to Staelin.

Claims 48 and 52 stand rejected under 35 U.S.C. § 103(a) as being obvious over Rowley in view of Swank and Iannucci.

1           Claims 58 and 59 stand rejected under 35 U.S.C. § 103(a) as being obvious  
2 over Rowley in view of Iannucci, Pacheco and an article entitled “The Component  
3 Object Model: A Technical Overview” by Williams (hereinafter “Williams”).

4           Claims 60 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
5 Rowley in view of Iannucci, Pacheco and Collins III.

6           Claims 64, 65, 67 and 68 stand rejected under 35 U.S.C. § 103(a) as being  
7 obvious over Collins III in view of U.S. Patent No. 5,859,973 to Carpenter.

8           Claim 66 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
9 Collins III in view of Carpenter and U.S. Patent No. 5,826,265 to Van Huben.

10          Claim 69 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
11 U.S. Patent No. 6,282,711 to Halpern in view of U.S. Patent No. 5,721,824 to  
12 Taylor.

13          Claim 70 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
14 Halpern in view of Taylor and Rowley.

15          Claims 77-80 stand rejected under 35 U.S.C. § 103(a) as being obvious  
16 over Rowley in view of Iannucci, Pacheco and U.S. Patent No. 4,910,663 to  
17 Bailey.

18          Claim 81 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
19 Rowley in view of Iannucci, Pacheco and U.S. Patent No. 5,761,408 to Kolawa.

20          Claims 83 and 85 stand rejected under 35 U.S.C. § 103(a) as being obvious  
21 over Rowley.

22          Claim 84 stands rejected under 35 U.S.C. § 103(a) as being obvious over  
23 Rowley in view of Bailey.

24          Claims 86-90 stand rejected under 35 U.S.C. § 103(a) as being obvious  
25 over Bailey in view of Kolawa and Collins III.

1 Claims 91-93, 95 and 96 stand rejected under 35 U.S.C. § 103(a) as being  
2 obvious over Rowley in view of Collins III and Bailey.

3 Before discussing the substance of the Office's rejections, the following  
4 discussion is provided to assist the Office in appreciating the patentable  
5 distinctions between Applicant's claimed subject matter and the various references  
6 cited by the Office.

### 7 8 **The Rowley Reference**

9 Rowley's is perhaps best appreciated with respect to FIG. 1, which shows a  
10 computer network comprising a number of client computers 101 and a number of  
11 server computers 102 interconnected by a network 103.

12 Each file server 102 stores a number of application files 104, forming a  
13 number of software applications. Normally, each application consists of several  
14 application files. The application files are stored in compressed form, using any  
15 standard data compression technique.

16 Conveniently, the server has a number of application directories, one for  
17 each application. Each of these directories has a number of sub-directories, which  
18 hold the new or amended application files for different versions of the application.  
19 Typically, one of these versions is an installer version of the application, while the  
20 other versions are non-installers. An installer is an executable program which,  
21 when run on a client machine, sets up the correct environment for the application.  
22 Normally, the first release of an application is an installer, and subsequent releases  
23 are non-installers.

24 Each file server also stores a number of manifest files 105, one for each  
25 version of each application stored on the server. These manifest files are stored in

1 the relevant sub-directories, and each has a name constructed from the name and  
2 release number of the application to which it relates. Each manifest file contains a  
3 list of the application files that make up the particular version of the application.  
4 For each application file, it contains the following parameters:

- 5 • The filename of the application file.
- 6 • The version number of the application file.
- 7 • The target directory into which the application file should be  
8 installed.
- 9 • Date and time of issue.
- 10 • File size and compressed file size.
- 11 • An action parameter which indicates whether the file is to be  
12 installed, to be deleted, or to be executed on download.
- 13 • A flag which indicates access permissions of the file, e.g. read only.
- 14 • A cyclic redundancy checksum (CRC).

15 FIGS. 3A and 3B show the operation of an update program 110 (Fig. 1).  
16 Referring to FIG. 3A, the update program first contacts one of the servers 102  
17 (Step 301), by way of the network 103, to obtain the live release file from that  
18 server. The update program then compares this release file with its locally held  
19 registration file 109 (Step 302), to identify which of the currently installed  
20 applications have more recent versions available. It also identifies any new  
21 application installers in the release file for applications that are not installed  
22 locally on the client.

23 The update program then displays a screen, as shown in FIG. 9, which  
24 allows the user to select either an "Updates" option or a "New Release" option. If  
25 the user selects the "Updates" option, the update program proceeds to step 303 in  
FIG. 3A. Alternatively, if the user selects the "New Release" option, the update  
program proceeds to step 310 in FIG. 3B.

1        If the user selects the "Updates" option (Step 303), a list of the titles and  
2 versions of currently installed applications is displayed, as shown in FIG. 9. The  
3 display indicates which, if any, of the applications have more recent versions  
4 available. The user may select from this list one or more (or all) of the  
5 applications for which a more recent version is available. Selection of an  
6 application will automatically cause any dependent applications to be selected.

7        If there is a more recent version available of the update program itself, this  
8 is automatically selected. Hence, every time the update program runs it will update  
9 itself if necessary.

10       If the user selects the "OK" button on this screen, the program proceeds to  
11 Step 304 below. Alternatively, the user may simply exit from the program without  
12 performing any updates by selecting the "Cancel" button. Assuming the "OK"  
13 button was selected in step 303, the update program contacts the server 102 to  
14 obtain the manifest file for the first (or only) of the selected applications (Step  
15 304). The update program then determines differences between files installed on  
16 the client computer and those listed in the manifest file (Step 305). For each  
17 application file listed in the manifest file, a check is made to determine whether  
18 the specified file is already present in the specified directory in the client by using  
19 CRC checks. If not, the program contacts the server 102, to retrieve the required  
20 application file. The retrieved file is expanded, and then checked for file-transfer  
21 corruption, using the CRC checksum. All the application files are read into a  
22 temporary directory on the client computer. Thus, the update program does not  
23 fetch any application file if the required version of that file is already installed in  
24 the required directory, thereby eliminating unnecessary traffic over the network.  
25

1 When all the files listed in the manifest file have been correctly retrieved  
2 (Step 306), the installation actions are implemented as follows. Any files marked  
3 for deletion are deleted from the client computer, files marked for execution are  
4 executed and files marked for installation are installed into the specified  
5 directories in the client, provided the file version is more advanced than that of the  
6 existing file. Hence then any existing files with the same names in the directories  
7 will be overwritten.

8 If, on the other hand, some of the file transfers failed, none of the files are  
9 installed. Instead, a message is displayed, giving the user the option of either  
10 canceling the update, or making another attempt to access the files.

11 If all the required applications have now been updated (Step 307), the  
12 update program proceeds to Step 308. Otherwise it returns to Step 304 above to  
13 get the manifest file for the next required application to be updated.

### 14 15 **The Iannucci Reference**

16 Iannucci discloses a system for sending and receiving automatic message  
17 notification and remote client configuration. As shown in Iannucci's Fig. 1, a  
18 server 100, a client 110 and at least one message server 120 communicate with  
19 each other via a communication network 130.

20 The server 100 stores a software application 159 and a database 155. The  
21 database 155 has a current version number 158 of a particular software  
22 application, which will be called "application A" and which may be stored as part  
23 of the programming 159, a list of clients 157, and information 160 relating to the  
24 availability of a message including persistent state information.

1       The client 110 includes a processor 170 having an elapsed time counter  
2       171, a display 180 and a memory 185. The memory 185 stores a database 190 and  
3       software 187, including the version of software “application A” which presently  
4       operates on the processor 170. The client 110 communicates with the server 100  
5       via the communications network 130 and is capable of directing the display of  
6       web pages 195 and 197 on the display 180.

7       An accept/reject button 191, a message waiting indicator 192, and a  
8       configuration or upgrade message 193 and client state change directives 198 are  
9       shown displayed within the web page 195 on the display 180. The button 191,  
10      configuration message 193, and the directives 198 are all part of the configuration  
11      information 196 which appears within web page 195. The message 196 appears in  
12      a separate web page 197.

13      The database 190 contains server Universal Resource Locators (URL's)  
14      including the URL 210 associated with the server 100, a first persistent state value  
15      A 230 and a second persistent state value B 240, message URL's including the  
16      URL 250 associated with the message server 120 and a frequency time 260. If  
17      available, the frequency time is a user selected minimum time period between  
18      upgrade availability notifications.

19      Referring additionally now to FIG. 2, in response to a client processor 170  
20      initializing the “application A” software stored on memory 185, the server 100, via  
21      the communication network 130, automatically receives a signal in step 300 which  
22      represents status information from the client 110, in accordance with programmed  
23      instructions included in the software 187 stored on the client memory 185. Among  
24      other information, the status information includes a client version number of the  
25      software “application A” stored on memory 185. Software “application A” may



1 have been originally downloaded to the client 110 from server 100, or from a  
2 different network server. Alternatively, software “application A” could have been  
3 loaded directly to the client 110 from a floppy disk or other physical storage  
4 medium.

5 The server 100, in step 320, compares the client version number of software  
6 “application A” stored in client 110 against a current version number 158 of the  
7 software “application A” stored in the memory 155 of server 100 and determines  
8 whether the current version number 158 is greater than the client version number.  
9 If so, the server processor 140, in accordance with programmed instructions which  
10 form part of the software 159 stored on server memory 105, generates  
11 configuration information 196 in step 330. The configuration information 196  
12 includes a configuration message 193 which contains information pertaining to  
13 features of the available upgrade of the software “application A” to current version  
14 number 158, and an offer to download the software application A upgrade. The  
15 configuration information 196 also includes, an accept/reject button 191. The  
16 configuration information 196 may further include client persistent state change  
17 directives 198. The server processor 140, in accordance with its programmed  
18 instructions, also directs the transmission of the configuration information 196 via  
19 the network 130 to the client 110 in step 330. The server processor 140, in step  
20 350, additionally determines whether the offer is accepted or rejected through the  
21 receipt of a new request from client 110 to download the upgrade. If the offer is  
22 accepted the server 100 first downloads a web page containing a link to the  
23 software upgrade. Responsive to the user clicking on the link, the server 100  
24 downloads the new version of the software to client 110 in step 360.  
25

1       The configuration information 196, including the configuration message  
2 193 and the accept/reject button 191 are displayed on the client display 180 in step  
3 460 of FIG. 3. By clicking on the displayed accept/reject button 191 to accept the  
4 upgrade of software “application A”, a signal is generated and communicated from  
5 the client processor 170 to the server processor 140 via the network 130,  
6 responsive to which the server processor 140, in step 360, directs the downloading  
7 of the web page containing the link to the upgrade to the client 110. If accepted,  
8 the upgrade is downloaded and stored by processor 170 on client memory 185. By  
9 clicking on the accept/reject button 191 to accept or reject the upgrade, the  
10 configuration information 196 will be eliminated from the display 180 as indicated  
11 in step 370. Until the button 191 is clicked on, the configuration information 196  
12 will continue to be displayed during the current session and will be redisplayed  
13 during each future session, subject to the selected minimum time periods between  
14 upgrade notification. If the upgrade is accepted or rejected, the previously  
15 displayed configuration information 196 will not be displayed during future  
16 sessions.

17  
18       **Claims Rejected under § 102 over Rowley and Related Dependent**  
19       **Claims Rejected Under § 103**

20       **Claim 82** recites a method for creating a package manifest comprising:

- 21       • receiving information pertaining to one or more extension directories  
22       that contain files that comprise a software extension that is to be  
23       incorporated into a software application program to extend the  
24       application program;  
25       • receiving, if any, information pertaining to file groups or load  
      dependencies;  
      • receiving, if any, information pertaining to file usage statistics; and  
      • generating a package manifest based on the received information.

1  
2 In making out the rejection of this claim, the Office argues that Rowley  
3 anticipates this claim's subject matter. Specifically, the Office cites to Rowley's  
4 column 4, lines 57-61 and Fig. 8 as disclosing subject matter that anticipates this  
5 claim. Applicant respectfully disagrees and traverses the rejection.

6 For example, the Office argues that Rowley discloses receiving information  
7 that pertains to file usage statistics and cites to Fig. 8 and argues that Rowley's  
8 flag indicating permissions of the file is a "file usage statistic." In addition, in  
9 addressing Applicant's previous arguments, the Office states that a "read only"  
10 flag *is* a statistic because it dictates *how* a file is used. See, e.g. Office Action,  
11 page 20, first full paragraph.

12 Applicant respectfully disagrees. Rowley's flag is provided to mark a  
13 particular file as a "read only" file. Applicant submits that there is nothing  
14 *statistical* about a flag that marks a file as a "read only" file as the term "file usage  
15 statistic" is used in the Specification.

16 As an example, consider the following material from the Specification  
17 starting on page 28, line 20 through page 30, line 6:

18  
19 The *file usage statistics* from scenario runs parameter...enables the  
20 file download priority to be determined based on scenario runs. A scenario  
21 is a script of tasks that the average user typically follows when using a  
22 product during a particular portion of product use. For example, one  
23 scenario might pertain to the tasks involved in sending an email message  
24 (i.e. click "new mail" button, type in "TO" well, type is "Subject" well,  
25 etc.). In the described embodiment, file usage statistics from scenario runs  
are collected from running IIS logs on various scenarios. *The different  
scenarios are directed to ensuring, with some degree of probabilistic  
support, that the file download order reflects, in some way, the files that  
will likely be used by the user first.*

1  
2 Hence, it should be quite clear that Rowley's "read only" flag neither  
3 discloses nor suggests file usage statistics as that term is utilized in the claim and  
4 the Specification. Accordingly, for at least this reason, this claim is allowable.

5 **Claims 83-85** depend from claim 82 and are allowable as depending from  
6 an allowable base claim. These claims are also allowable for their own recited  
7 features which, in combination with those recited in claim 82, are neither disclosed  
8 nor suggested in the references of record, either singly or in combination with one  
9 another. In addition, given the allowability of claim 82, the rejection of claim 84  
10 over the combination with Bailey is not seen to add anything of significance.

### 11 12 **The § 103 Standard**

13 To establish a prima facie case of obviousness, three basic criteria *must* be  
14 met. First, there must be some suggestion or motivation, either in the references  
15 themselves or in the knowledge generally available to one of ordinary skill in the  
16 art, to modify the reference or to combine reference teachings. *In re Jones*, 958  
17 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992); *In re Fine*, 837 F.2d 1071, 5  
18 USPQ2d 1596 (Fed. Cir. 1988). Second, there must be a reasonable expectation  
19 of success. *In re Merck & Co., Inc.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir.  
20 1986). Finally, the prior art reference (or references when combined) must teach  
21 or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580  
22 (CCPA 1974).

23 Hence, when patentability turns on the question of obviousness, the search  
24 for and analysis of the prior art includes evidence relevant to the finding of  
25 whether there is a teaching, motivation, or suggestion to select and combine the

1 references relied on as evidence of obviousness. See, e.g., *McGinley v. Franklin*  
2 *Sports, Inc.*, 262 F.3d 1339, 1351-52, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001)  
3 ("the central question is whether there is reason to combine [the] references," a  
4 question of fact drawing on the Graham factors).

5 "The factual inquiry whether to combine references must be thorough and  
6 searching." *Id.* It must be based on objective evidence of record. This precedent  
7 has been reinforced in myriad decisions, and cannot be dispensed with. See, e.g.,  
8 *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-  
9 25, 56 USPQ2d 1456, 1459 (Fed. Cir. 2000) ("a showing of a suggestion,  
10 teaching, or motivation to combine the prior art references is an 'essential  
11 component of an obviousness holding'" (quoting *C.R. Bard, Inc. v. M3 Systems,*  
12 *Inc.*, 157 F.3d 1340, 1352, 48 USPQ2d 1225, 1232 (Fed. Cir. 1998)); *In re*  
13 *Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999) ("Our  
14 case law makes clear that the best defense against the subtle but powerful  
15 attraction of a hindsight-based obviousness analysis is rigorous application of the  
16 requirement for a showing of the teaching or motivation to combine prior art  
17 references."); *In re Dance*, 160 F.3d 1339, 1343, 48 USPQ2d 1635, 1637 (Fed.  
18 Cir. 1998) (there must be some motivation, suggestion, or teaching of the  
19 desirability of making the specific combination that was made by the applicant); *In*  
20 *re Fine*, 837 F.2d 1071, 1075, 5 USPQ2d 1596, 1600 (Fed. Cir. 1988) ("teachings  
21 of references can be combined only if there is some suggestion or incentive to do  
22 so.") (emphasis in original) (quoting *ACS Hosp. Sys., Inc. v. Montefiore Hosp.*,  
23 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed. Cir. 1984)).

24 The need for specificity pervades this authority. See, e.g., *In re Kotzab*, 217  
25 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000) ("particular findings

1 must be made as to the reason the skilled artisan, with no knowledge of the  
2 claimed invention, would have selected these components for combination in the  
3 manner claimed").

4  
5 **Claims Rejected Over the Combination that Includes at least Rowley  
and Iannucci**

6 **Claim 1** has been amended and recites a method for delivering software via  
7 a network comprising [added language appears in bold italics]:

- 8
- 9 • describing one or more software extensions using a hierarchical  
10 language, the extensions being configured for incorporation on a  
client, said describing defining one or more manifests containing at  
least one list of files comprising an extension; and
  - 11 • delivering the one or more manifests to the client via the network,  
12 the one or more manifests being configured for use in downloading  
the software extensions via the network, at least some of the  
13 extensions being downloadable by streaming extension files to the  
client in a manner that enables a user to begin to interact with the  
14 extension sooner than if the user had to wait for the entire extension  
to load, *said manner being developed based on scenario runs in  
15 which files that are more likely to be first used by the user are  
downloaded before files that are less likely to be first used, and  
16 wherein files that are less likely to be used first can be downloaded  
via a background download process.*
- 17

18

19 In making out the rejection of this claim, the Office argues that Rowley  
20 teaches the recited acts of describing and delivering, except for describing the  
21 extensions using a hierarchical language. The Office then relies on Iannucci and  
22 argues that it teaches a manifest file in the form of a web page which is sent to the  
23 client for use in downloading a software extension. In addition, responsive to the  
24 previous amendment, the Office has added Pacheco to the combination and argues  
25

1 that Pacheco teaches downloading “any type” of data to a client using streaming—  
2 which allows users to interact with the files while the file is downloading.

3 Based on this, the Office argues that this claim would be obvious in view of  
4 the combination of Rowley, Iannucci and Pacheco.

5 Applicant respectfully disagrees and maintains its position with respect to  
6 the Office’s interpretation and application of Iannucci. Nonetheless, Applicant has  
7 amended claim 1 to recite that the “manner” that enables a user to begin to interact  
8 with the extension sooner than if the user had to wait for the entire extension to  
9 load is *developed based on scenario runs in which files that are more likely to be*  
10 *first used by the user are downloaded before files that are less likely to be first*  
11 *used, and wherein files that are less likely to be used first can be downloaded via*  
12 *a background download process*. Support for this amendment can be found in the  
13 Specification on page 18, lines 18-24.

14 Pacheco, on which the Office relies for its streaming teaching, generally  
15 discusses the notion of data streaming. Pacheco does not teach or suggest scenario  
16 runs in which files that are more likely to be first used by a user are downloaded  
17 before other files. Further, Pacheco does not teach or suggest downloading other  
18 such files via a background download process.

19 Accordingly, for at least this reason, claim 1 is allowable.

20 **Claims 2-12** depend from claim 1 and are allowable as depending from an  
21 allowable base claim. These claims are also allowable for their own recited  
22 features which, in combination with those recited in claim 1, are neither disclosed  
23 nor suggested in the references of record, either singly or in combination with one  
24 another. In addition, given the allowability of claim 1 over the combination of  
25 Rowley and Iannucci, the rejections of claim 6 over the further combination with

1 Swank, and of claim 7 over the further combination with Miller are not seen to add  
2 anything of significance.

3 **Claim 13** recites one or more computer-readable media comprising  
4 computer-readable instructions thereon which, when executed by a computer,  
5 cause the computer to [added language appears in bold italics]:

- 6
- 7 • describe one or more software extensions using extensible markup  
8 language (XML), the extensions being configured for incorporation  
9 on a client, said describing defining a manifest containing at least  
10 one list of files comprising an extension, the manifest being  
11 configured to assist in one or more of the following: organizing  
12 delivery of individual files listed in the manifest, validating  
13 individual files listed in the manifest, and updating individual files  
14 listed in the manifest; and
- 15 • deliver the manifest to the client via the network, at least some of the  
16 extensions being downloadable by streaming extension files to the  
17 client in a manner that enables a user to begin to interact with the  
18 extension sooner than if the user had to wait for the entire extension  
19 to load, *said manner being developed based on scenario runs in  
20 which files that are more likely to be first used by the user are  
21 downloaded before files that are less likely to be first used, and  
22 wherein files that are less likely to be used first can be downloaded  
23 via a background download process.*
- 24
- 25

18 In making out the rejection of this claim, the Office argues that the  
19 combination of Rowley, Pacheco and Iannucci render the subject matter of this  
20 claim obvious. Applicant respectfully disagrees with the Office's combination  
21 and rationale. Nonetheless, Applicant has amended this claim to recite that the  
22 "manner" that enables a user to begin to interact with the extension sooner than if  
23 the user had to wait for the entire extension to load is *developed based on scenario  
24 runs in which files that are more likely to be first used by the user are  
25 downloaded before files that are less likely to be first used, and wherein files that*



1 *are less likely to be used first can be downloaded via a background download*  
2 *process.* Support for this amendment can be found in the Specification on page  
3 18, lines 18-24.

4 Pacheco, on which the Office relies for its streaming teaching, generally  
5 discusses the notion of data streaming. Pacheco does not teach or suggest scenario  
6 runs in which files that are more likely to be first used by a user are downloaded  
7 before other files. Further, Pacheco does not teach or suggest downloading other  
8 such files via a background download process.

9 Accordingly, for at least this reason, this claim is allowable.

10 **Claim 14** has been amended and recites a method for receiving software  
11 via a network comprising [added language appears in bold italics]:

- 12  
13 • receiving a manifest that contains at least one list of files comprising  
14 a software extension that is to be downloaded via a network and  
incorporated on a client, the manifest being defined in extensible  
15 markup language (XML), the manifest being configured to assist in:
  - 16 ○ organizing delivery of the files,
  - 17 ○ validating individual files listed in the manifest, and
  - 18 ○ updating individual files listed in the manifest; and
  - 19 ○ downloading files from the list of files contained in the  
20 manifest;
- 21 • wherein the extension is downloadable by streaming extension files  
22 to the client in a manner that enables a user to begin to interact with  
the extension sooner than if the user had to wait for the entire  
23 extension to load, *said manner being developed based on scenario*  
*runs in which files that are more likely to be first used by the user*  
*are downloaded before files that are less likely to be first used, and*  
*wherein files that are less likely to be used first can be downloaded*  
*via a background download process.*

24 In making out the rejection of this claim, the Office makes the same argues  
25 that it made with respect to claim 13 above. Applicant disagrees with the Office's

1 rejection and rationale. Nonetheless, Applicant has amended this claim to recite  
2 that the “manner” that enables a user to begin to interact with the extension sooner  
3 than if the user had to wait for the entire extension to load is *developed based on*  
4 *scenario runs in which files that are more likely to be first used by the user are*  
5 *downloaded before files that are less likely to be first used, and wherein files that*  
6 *are less likely to be used first can be downloaded via a background download*  
7 *process*. Support for this amendment can be found in the Specification on page  
8 18, lines 18-24.

9 Pacheco, on which the Office relies for its streaming teaching, generally  
10 discusses the notion of data streaming. Pacheco does not teach or suggest scenario  
11 runs in which files that are more likely to be first used by a user are downloaded  
12 before other files. Further, Pacheco does not teach or suggest downloading other  
13 such files via a background download process.

14 Accordingly, for at least this reason, this claim is allowable.

15 **Claims 15-20** depend from claim 14 and are allowable as depending from  
16 an allowable base claim. These claims are also allowable for their own recited  
17 features which, in combination with those recited in claim 14, are neither disclosed  
18 nor suggested in the references of record, either singly or in combination with one  
19 another.

20 **Claim 37** has been amended and recites a method of providing software via  
21 a network comprising [added language appears in bold italics]:

- 22
- 23 • describing one or more software extensions using one or more  
24 extensible markup language (XML) files, the extensions being  
25 configured for incorporation in a software program executing on a  
client, individual XML files providing individual manifests that  
contain a list of files that comprise an extension; and

- storing the XML files in a Web-accessible location;
- wherein at least some of the extensions are downloadable by streaming extension files to the client in a manner that enables a user to begin to interact with the extension sooner than if the user had to wait for the entire extension to load, *said manner being developed based on scenario runs in which files that are more likely to be first used by the user are downloaded before files that are less likely to be first used, and wherein files that are less likely to be used first can be downloaded via a background download process.*

In making out the rejection of this claim, the Office argues that the combination of Rowley, Pacheco and Iannucci renders the subject matter of this claim obvious. Applicant respectfully disagrees with the Office's combination and rationale. Nonetheless, Applicant has amended this claim to recite that the "manner" that enables a user to begin to interact with the extension sooner than if the user had to wait for the entire extension to load is *developed based on scenario runs in which files that are more likely to be first used by the user are downloaded before files that are less likely to be first used, and wherein files that are less likely to be used first can be downloaded via a background download process.* Support for this amendment can be found in the Specification on page 18, lines 18-24.

Pacheco, on which the Office relies for its streaming teaching, generally discusses the notion of data streaming. Pacheco does not teach or suggest scenario runs in which files that are more likely to be first used by a user are downloaded before other files. Further, Pacheco does not teach or suggest downloading other such files via a background download process.

Accordingly, for at least this reason, this claim is allowable.

1       **Claims 38-46** depend from claim 37 and are allowable as depending from  
2 an allowable base claim. These claims are also allowable for their own recited  
3 features which, in combination with those recited in claim 37, are neither disclosed  
4 nor suggested in the references of record, either singly or in combination with one  
5 another. Additionally, given the allowability of claim 37, the rejections of claims  
6 40 and 41 over the further combination with Swank is not seen to add anything of  
7 significance.

8       **Claim 55** recites a data structure embodied on a computer-readable  
9 medium comprising:

- 10       • one or more first tags indicative of associated file groups associated  
11       with an Internet-downloadable software extension that can extend an  
12       application program executing on a client; and
- 13       • one or more second tags indicative of specific files that comprise the  
14       software extension.

15       In making out the rejection of this claim, the Office argues that Rowley  
16 teaches a manifest file which stores a list of files utilized in a software extension  
17 and further teaches one or more files groups associated with the files. The Office  
18 admits that Rowley does not teach that the individual files and file groups  
19 comprise tags that indicate individual files and file groups. The Office then relies  
20 on Iannucci and argues that it teaches a manifest in the form of a web page, which  
21 is sent to the client for use in downloading a software extension. The Office then  
22 reasons that web pages are designed using HTML—a tag-based language—and  
23 that using tags to separate fields in a web page is an inherent aspect of HTML.  
24 Based on this, the Office argues that the subject matter of this claim is obvious in  
25

1 view of Rowley and Iannucci since the combination allows the manifest to take  
2 the form of a web page.

3 In addition, the Office addresses Applicant's previous arguments traversing  
4 this rejection on page 18, last paragraph of the present Office Action. Specifically,  
5 the Office argues that Rowley teaches that files are separated into specific  
6 directories and target directories which are file groups.

7 Applicant respectfully disagrees. First, the portion of Iannucci cited by the  
8 Office in support of this rejection pertains simply to a web page with a *clickable*  
9 *link* that enables a user to click on the link and subsequently download software.  
10 Second, and perhaps more importantly, there is no disclosure in either reference to  
11 provide a data structure that associates one or more first tags with *file groups* and  
12 one or more second tags with *specific files*. Rather, Rowley's manifest appears to  
13 simply list application files that make up a particular version of an application.  
14 More specifically, the parameters that are included in Rowley's manifest include a  
15 file name of an application file, a version number, a target directory, a date and  
16 time of issue, and additional information—none of which is disclosed to comprise  
17 file groups. Additionally, Iannucci simply discloses a *clickable link*. There is no  
18 teaching or suggestion that such link is defined by or defines *both* first tags  
19 indicative of associated file groups and second tags indicative of specific files that  
20 comprise a software extension.

21 For at least these two reasons, the Office has failed to establish a *prima*  
22 *facie* case of obviousness and this claim is allowable.

23 **Claims 56-61** depend from claim 55 and are allowable as depending from  
24 an allowable base claim. These claims are also allowable for their own recited  
25 features which, in combination with those recited in claim 55, are neither disclosed

1 nor suggested in the references of record, either singly or in combination with one  
2 another. In addition, given the allowability of claim 55, the rejections of claims  
3 58-59 over Williams, and of claim 60 over Collins, is not seen to add anything of  
4 significance.

5 **Claim 71** has been amended and recites an automated software tool  
6 comprising a package manifest creation tool configured to [added language  
7 appears in bold italics]:

- 8
- 9 • receive one or more input parameters pertaining to a package  
10 manifest that is to describe a software extension that is configured to  
11 extend a software application executing on a client; and
- 12 • generate a package manifest that describes the extension, the  
13 package manifest being generated using a hierarchical language;
- 14 • wherein the extension is downloadable by streaming extension files  
15 to the client in a manner that enables a user to begin to interact with  
16 the extension sooner than if the user had to wait for the entire  
17 extension to load, *said manner being developed based on scenario  
18 runs in which files that are more likely to be first used by the user  
19 are downloaded before files that are less likely to be first used, and  
20 wherein files that are less likely to be used first can be downloaded  
21 via a background download process.*
- 22
- 23
- 24
- 25

18 In making out the rejection of this claim, the Office argues that Rowley  
19 teaches a manifest editor (Fig. 8) as claimed, but fails to teach that the manifest is  
20 described using a hierarchical language. The Office then relies on Iannucci to  
21 argue that the subject matter of this claim is obvious. In addition, the Office  
22 admits that neither Rowley nor Iannucci teach that extensions are downloadable  
23 by streaming the extension files to the client in a manner that enables a user to  
24 begin to interact with the extension sooner than if the user had to wait for the  
25

1 entire extension to load. To supply this subject matter, the Office relies on  
2 Pacheco.

3 Applicant respectfully disagrees with the Office's combination and  
4 rationale. Nonetheless, Applicant has amended this claim to recite that the  
5 "manner" that enables a user to begin to interact with the extension sooner than if  
6 the user had to wait for the entire extension to load is *developed based on scenario*  
7 *runs in which files that are more likely to be first used by the user are*  
8 *downloaded before files that are less likely to be first used, and wherein files that*  
9 *are less likely to be used first can be downloaded via a background download*  
10 *process*. Support for this amendment can be found in the Specification on page  
11 18, lines 18-24.

12 Pacheco, on which the Office relies for its streaming teaching, generally  
13 discusses the notion of data streaming. Pacheco does not teach or suggest scenario  
14 runs in which files that are more likely to be first used by a user are downloaded  
15 before other files. Further, Pacheco does not teach or suggest downloading other  
16 such files via a background download process.

17 Accordingly, for at least this reason, this claim is allowable.

18 **Claims 72-81** depend from claim 71 and are allowable as depending from  
19 an allowable base claim. These claims are also allowable for their own recited  
20 features which, in combination with those recited in claim 71, are neither disclosed  
21 nor suggested in the references of record, either singly or in combination with one  
22 another. In addition, given the allowability of claim 71, the rejections of claims 77  
23 and 81 over the references to Bailey and Kolawa respectively, are not seen to add  
24 anything of significance.  
25

1                   **Claims Rejected Over the Combination that Includes at least Rowley**  
2                   **and Swank**

3                   **Claim 21** recites a data structure comprising:

- 4
- 5                   • a list of one or more files that are utilized in a software extension  
6                   that is configured to extend a software application executing on a  
7                   client;
  - 8                   • one or more hashes each of which being associated with a particular  
9                   listed file; and
  - 10                  • one or more file groups, individual files being associated with  
11                  individual file groups, the file groups determining when particular  
12                  files of the extension get downloaded to the client;
  - 13                  • *the data structure being configured to assist in delivering software  
14                  extensions via the Internet.*

15                  In making out the rejection of this claim, the Office argues that Rowley  
16                  teaches a manifest file that stores a list of files and further teaches one or more file  
17                  groups associated with files. The Office further argues that Rowley teaches  
18                  selecting certain groups of files for downloading to form an application program  
19                  and hence, determining when the files are downloaded based on which groups the  
20                  files are in, since selecting a group of files determines that the groups of files is  
21                  downloaded to the client.

22                  The Office attempts to clarify its argument in its “Response to Arguments”  
23                  section on page 18, first full paragraph. Specifically, the Office notes that it is true  
24                  that Rowley’s file groups determine *what* particular files of the extension get  
25                  downloaded. The Office then extends this by arguing that Rowley’s file groups  
allow users to choose file groups to download and hence, *when* the groups get  
downloaded. This logic is misplaced for at least the following reason. In Rowley,  
it is the *user* who determines when the files are downloaded. *This claim*



1 *specifically recites that the file groups determine when particular files of the*  
2 *extension get downloaded.* This is not to state or imply that the user is not  
3 involved in the process. Quite to the contrary, Applicant's disclosure provides  
4 examples of situations in which the user is involved in the download process.  
5 What this claim recites, though, is that the file groups determine *when* particular  
6 files get downloaded.

7 The Office is respectfully referred to Applicant's Specification, starting on  
8 page 19, line 15 where an exemplary implementation of a file group is described.  
9 The discussion appearing in this portion of the Specification is provided below for  
10 the Office's convenience:

11  
12 All files in an extension can be labeled according to a number  
13 of predefined file groups. The file group of a particular file  
14 determines *when the particular file gets downloaded*, where it is  
15 stored on the client, and how it gets packaged. In the described  
16 embodiment, four predefined file groups are provided and are listed  
17 and described in the table immediately below:  
18  
19  
20  
21  
22  
23  
24  
25

Group name	When downloaded	Where stored on the client	Packaging	Content
Required	Downloaded before any other files in the extension.	NetDocs package cache	All required files in an extension are packaged together as a CAB* file.	DLLs included so that a user will not have to wait for a prolonged period of time before clicking on a UI element
Offline	Offline files start getting downloaded as soon as Required are down. Providing the user stays on line long enough, these files will all get downloaded and will later be available for offline use.	NetDocs package cache	File are sent down individually.	Bulk of the UI files.
On demand	Only downloaded when they are requested for the first time.	NetDocs package cache	Files are sent down individually.	To avoid using up disk space on the client, advanced features can be put in this category.
Online only	Downloaded on demand. Content is only available when the user is online.	WinInet Cache	Files are sent down individually	Content that is not to be provided offline. Examples include help pages and other content that can consume a large amount of disk space.

The Office will notice in this example, that the table describes four file group names which appear in the first column. Additionally, the second column describes when files of the group are to be downloaded. For example, "Required" files get downloaded before any other files in the extension; and "Offline" files start getting downloaded as soon as file in the "Required" group are downloaded.

Rowley neither discloses nor suggests any such subject matter. Accordingly, for at least this reason, the Office has failed to establish a *prima facie* case of obviousness and this claim is allowable.

The Office then relies on Swank, which teaches storing a hash for a file, to argue that the subject matter of this claim would be obvious. Applicant respectfully disagrees. In clarifying its position and addressing the Applicant's previous arguments, the Office states that "Swank is directed to downloading data from a server to a client using a hash to determine changes in files to determine what to download to a client." The Office then states that "[t]his is comparable to

1 downloading extensions from a server to a client using a hash to determine which  
2 files to download to the client.” Applicant respectfully disagrees—*comparability*  
3 is not the standard that is to be used to make out an obviousness rejection.

4 As previously noted, Swank has nothing whatsoever to do with delivering  
5 software extensions via the Internet. Rather, Swank is directed to minimizing  
6 communications between a host computer and a personal computer by transmitting  
7 only changed lines in an updated file from a terminal to a host. Swank uses its  
8 hash processing to ascertain which text lines have been changed and hence which  
9 text lines need to be transmitted. Swank does not disclose or suggest hashes in the  
10 context of a data structure that comprises the other components recited above *to*  
11 *assist in delivering software extensions via the Internet.*

12 It would appear that the Office’s attempted combination is, at best, based  
13 on hindsight reconstruction which has been specifically proscribed by the Federal  
14 Circuit. Accordingly, for at least these additional reasons, the Office has failed to  
15 establish a *prima facie* case of obviousness and this claim is allowable.

16 **Claims 22-36** depend from claim 21 and are allowable as depending from  
17 an allowable base claim. These claims are also allowable for their own recited  
18 features which, in combination with those recited in claim 21, are neither disclosed  
19 nor suggested in the references of record, either singly or in combination with one  
20 another. In addition, given the allowability of claim 21, the rejections of claims 25  
21 and 29-31 over Collins; of claim 26 over Mastrianni; and of claims 33-35 over  
22 Staelin is not seen to add anything of significance.

23 **Claim 47** recites a security method for downloading software extensions  
24 via the Internet comprising:  
25

- receiving, via the Internet, a package manifest containing a list of multiple files that comprise a software extension that is to be incorporated into an application program executing on a client, the list containing a hash for one or more of the files comprising the software extension;
- receiving, via the Internet, the multiple files that are described in the package manifest;
- creating a hash for one or more of the multiple received files; and
- comparing the created hash of the one or more files with corresponding file hashes contained in the package manifest to ascertain whether one or more of the received file is secure.

In making out the rejection of this claim, the Office essentially maintains its position as articulated in the previous Office Action dated October 28<sup>th</sup>, 2003. In that Office Action, the Office argues that Rowley teaches receiving a package manifest and multiple files as recited in this claim. The Office admits that Rowley does not teach that the list contains a hash for one or more of the files comprising the software extension, creating a hash for one or more received files and comparing the created hash with the corresponding file hashes to ascertain the security of the file.

The Office then relies on Swank and argues that it discloses storing a hash for an individual file to be updated, creating an updated hash of a received file, and comparing the hashes to determine the integrity of the file [sic:process], citing to column 3, lines 45-51. Based on this, the Office argues that the subject matter of this claim would be obvious in view of these two references. Applicant respectfully disagrees.

The Office further addresses Applicant's arguments against this combination on page 19, first full paragraph. Specifically, the Office argues that because Swank is directed to downloading data from a server to a client, this is

1 comparable to downloading extensions from a server to a client. Applicant  
2 submits that *comparability* is not the obviousness standard.

3 As noted above, Swank has nothing whatsoever to do with receiving  
4 software extensions via the Internet. Rather, Swank is directed to improving  
5 communications by transmitting only changed lines in an updated file from a  
6 terminal to a host. See, column 3, lines 24-30. Swank's hash process is very  
7 specifically directed to minimizing the amount of data that is sent between a  
8 terminal and a host. Specifically, as set forth in column 3, a text file together with  
9 its hash is first communicated from a host to a terminal. A non-editable hash file  
10 is created at the terminal and contains the host-computed hash value and a  
11 terminal-generated hash value for each line of the file. After some editing has  
12 occurred to the text file, a hash is re-computed for each line of text and compared  
13 line-by-line with the previously-computed has file to ascertain which lines have  
14 changed. Once identified, only changed lines are sent to the host. The only thing  
15 that Swank and this claim have in common is that the term "hash" is mentioned.  
16 Again, this is not the standard that is to be used in making out an obviousness  
17 rejection.

18 Accordingly, for at least this reason, this claim is allowable.

19 **Claims 48-50** depend from claim 47 and are allowable as depending from  
20 an allowable base claim. These claims are also allowable for their own recited  
21 features which, in combination with those recited in claim 47, are neither disclosed  
22 nor suggested in the references of record, either singly or in combination with one  
23 another. In addition, given the allowability of claim 47, the rejection of claim 48  
24 over Iannucci is not seen to add anything of significance.

1           **Claim 51** recites an updating method for updating software extensions via  
2 the Internet comprising:

- 3           • receiving, via the Internet, a package manifest containing a list of  
4           multiple files that comprise a newer version of a software extension  
5           that is to be incorporated into an application program executing on a  
6           client that contains an older software extension version, the list  
7           containing a hash for one or more of the files comprising the newer  
8           version of the software extension;
- 9           • comparing one or more hashes that are received with one or more  
10          hashes of files from the older version of the software extension;
- 11          • for any hashes of corresponding files from the different versions that  
12          are different, downloading a new file from a web server; and
- 13          • for any hashes of corresponding files from the different versions that  
14          are the same, copying a file from an old local directory on the client  
15          to a new local directory on the client associated with the newer  
16          version of the extension.

17           In making out the rejection of this claim, the Office argues that Rowley  
18 teaches receiving a package manifest and comparing files of an older version with  
19 files of a newer version stored in a manifest, and if corresponding files are  
20 different, downloading the new files. The Office admits that Rowley does not  
21 teach the use of hashes and relies instead on Swank. Relying on Swank, the  
22 Office argues that Swank teaches storing a file hash for a file that is to be updated,  
23 creating an updated hash of a received file and comparing the hashes so that when  
24 the hash of an old file and a newly received file is the same, the old file is updated  
25 and replaced by the new file.

          The Office further argues that it is possible to use Swank's method to  
determine which files need to be downloaded to the client. See, e.g. Office  
Action, page 10, second full paragraph. *Applicant respectfully submits that the*

1 possibility of using Swank in the manner argued by the Office is **not the legally**  
2 **correct obviousness standard.**

3 Applicant respectfully submits that the Office has misinterpreted this  
4 reference. Responsively, the Office argues that it uses Swank simply for the  
5 proposition of using hashes to determine changes in files.

6 According to Swank's system, an original file is maintained at the host and  
7 has an associated hash. When changes are made to the file on a remote terminal,  
8 those changes are communicated to the host. As specifically taught by Swank, the  
9 original file at the host is only updated after a hash value for the original entire file  
10 has been recomputed by the host and checked so that it matches the entire file hash  
11 total previously stored at the terminal. This ensures that the original file *at the*  
12 *host* has not been changed since the terminal hash file was generated. See, column  
13 3, lines 45-50. Thus, Swank teaches only using a hash to determine that an *old file*  
14 at the host is still, in fact, the *old file* from which the terminal's hash was  
15 computed. Once confirmed, only those changes transmitted by the terminal are  
16 made to the file. The terminal does not transmit the entire changed file—in fact,  
17 Swank's subject matter is specifically directed to avoiding any such situation.  
18 Thus, at best, Swank simply teaches using a hash to ascertain whether the host's  
19 old file has been changed since the terminal file's hash was computed so that the  
20 host can incorporate only those changes transmitted to the host by the terminal.

21 Recharacterized a bit, Swank's approach can be thought of as taking a hash  
22 of an old file and if the hash of the old file is the same as the hash of the old file at  
23 the terminal, making the changes sent by the terminal. Applicant's claim recites,  
24 *inter alia*, "for any hashes of corresponding files from the different versions that  
25

1 are different, downloading a new file from a web server”. Thus, this subject  
2 matter is directed to downloading a new file if the hashes are different.

3 It would appear that Swank is not on point for a couple of different reasons.  
4 First, as noted above, Swank is not at all directed to updating software extensions  
5 as recited in this claim. Rather, Swank appears to be directed to a non-analogous  
6 area of art. Second, Swank’s process relied upon by the Office is not directed to  
7 comparing hashes for newer versions of anything with older versions of anything  
8 for the purpose acquiring a newer version. Rather, Swank’s process is directed to  
9 comparing such hashes and, if different, possibly acquiring the older version (i.e.  
10 the old document) so that the text in the old document can be updated.

11 The Office’s attempted combination is misplaced and, at best, rests on  
12 hindsight reconstruction which has been specifically proscribed by the Federal  
13 Circuit.

14 For at least these reasons, the Office has failed to establish a *prima facie*  
15 case of obviousness and this claim is allowable.

16 **Claims 52-54** depend from claim 51 and are allowable as depending from  
17 an allowable base claim. These claims are also allowable for their own recited  
18 features which, in combination with those recited in claim 51, are neither disclosed  
19 nor suggested in the references of record, either singly or in combination with one  
20 another. In addition, given the allowability of claim 51, the rejection of claims 52  
21 and 54 over Iannucci is not seen to add anything of significance.

22  
23 **Claims Rejected Over the Combination that Includes at least Collins**  
24 **and Carpenter**

25 **Claim 64** recites a queue management method comprising:



- defining a download queue that controls when files are to be downloaded to a client, the files pertaining to a software extension that is to be incorporated into an application program executing on the client;
- ascertaining whether a user action at the client requires one or more files that are not currently being downloaded; and
- manipulating the download queue responsive to a user action that requires one or more files that are not currently being downloaded so that the one or more required files are downloaded sooner than they would otherwise be.

In making out the rejection of this claim, the Office argues that Collins teaches a download queue that controls when files are to be downloaded to a client. The Office admits that Collins does not teach ascertaining whether a user action at the client requires one or more files that are not currently being downloaded, and manipulating the download queue responsive to a user action that requires one or more files that are not currently being downloaded so that the one or more required files are downloaded sooner than they would otherwise be.

The Office then relies on Carpenter and argues that Carpenter teaches queue manipulation based on user input, citing to column 7, lines 21-25. Based on this, the Office argues that the subject matter of this claim would be obvious in view of Collins and Carpenter reasoning that such would allow a user to prioritize files that need to be downloaded while the queue is in progress. Further, the Office addresses Applicant's previous arguments with respect to Carpenter and notes that Carpenter was "introduced to show that a feature of a queue could be user manipulation of the queue". See, e.g. Office Action page 19 second full paragraph.

Applicant respectfully disagrees and submits that the Office has taken Carpenter out of context. Specifically, Carpenter's system and method are

1 designed to operate for transmitting messages from a memory constrained data  
2 processor to a host computer. See, e.g. column 4, lines 47-49. That is, in  
3 Carpenter, it is the client device that is memory constrained and transmission  
4 occurs *from* the client *to* the host. Thus, in Carpenter the queue management  
5 referred to by the Office (column 7, lines 21-25) takes place with respect to  
6 transmissions made from the client. Put another way, the queue management does  
7 not take place with respect to the client downloading anything. This is particularly  
8 germane when the claim language of this claim is examined.

9 Specifically, the claim recites that the download queue is defined and  
10 controls when files are to be *downloaded* to a client. The remainder of the claim  
11 (the recited acts of ascertaining and manipulating) are directed to ascertaining  
12 whether a user action requires one or more files that are not currently being  
13 downloaded and if so, manipulating the download queue so that one or more  
14 required files are downloaded sooner than they would otherwise be. Hence,  
15 Carpenter fails to teach queue management as recited in this claim. Specifically,  
16 Carpenter fails to teach the subject matter missing from Collins III—i.e.  
17 “ascertaining whether a user action at the client requires one or more files that are  
18 not currently being downloaded, and manipulating the download queue responsive  
19 to a user action that requires one or more files that are not currently being  
20 downloaded so that the one or more required files are downloaded sooner than  
21 they would otherwise be”. Carpenter’s scant teaching of user manipulation of a  
22 queue falls far short of supplying this missing subject matter and a motivation to  
23 use it to modify Collins III.

24 Accordingly, for at least this reason, the Office has failed to establish a  
25 *prima facie* case of obviousness and this claim is allowable.

1       **Claims 65-68** depend from claim 64 and are allowable as depending from  
2 an allowable base claim. These claims are also allowable for their own recited  
3 features which, in combination with those recited in claim 64, are neither disclosed  
4 nor suggested in the references of record, either singly or in combination with one  
5 another. In addition, given the allowability of claim 64, the rejection of claim 66  
6 over the combination with Van Huber is not seen to add anything of significance.

7  
8       **Claims Rejected Over the Combination of at least Halpern and Taylor**

9       **Claim 69** recites a method of creating software packages for delivery via  
10 the Internet comprising:

- 11
- 12       • identifying end user features;
  - 13       • identifying shared dependencies between the end user features;
  - 14       • creating individual software packages for the end user features;
  - 15       • creating individual software packages for the shared dependencies;
  - 16       and
  - 17       • hosting the software packages on a web server;
  - 18       • wherein both of said acts of identifying and both of said acts of
  - 19       creating provide software extensions that are created in a uniform
  - 20       manner independent of end user input.

21       In making out the rejection of this claim, the Office argues that Halpern  
22 teaches identifying end user features, creating individual software packages and  
23 hosting the software packages on a web server. The Office then relies on Taylor  
24 and argues that it teaches identifying shared dependencies between end user  
25 features and creating individual software packages for the shared dependencies.  
Based on the teachings of these two references, the Office argues that the subject

1 matter of this claim is obvious. Applicant previously disagreed with the Office's  
2 position and continues to disagree.

3         Nonetheless, Applicant previously made a clarifying amendment to this  
4 claim that both of the acts of identifying and both of the acts of creating provide  
5 software extensions that are created in a *uniform manner independent of end user*  
6 *input*. Specifically, the subject matter of this claim is directed to embodiments  
7 that enable packages to be created in a uniform manner in order to provide an  
8 organized delivery process. As noted in the Specification, one of the features of  
9 the described embodiment pertains to its extensibility. More specifically, software  
10 packages can be created by third party developers for extending so-called software  
11 platforms. See, e.g. Specification, page 49, lines 10-14 ("The embodiments  
12 described above provide a platform solution that provides for customization and  
13 extensibility through a consistent and logical extensibility mechanism and object  
14 model that can be easily understood by third party developers. Internet-based  
15 downloads can be accomplished without a great deal of user intervention and  
16 without manipulating any user persisted settings.").

17         Halpern, as Applicant previously noted, is very specifically directed to  
18 building installation packages that necessarily *require* user input. See, e.g.  
19 Abstract ("An installation package delivered to a requesting end user is custom  
20 configured at a remote server...in response to the user's inputs); column 5, lines  
21 49-51 ("In response to the user's selections, the options manager 104 delivers an  
22 installation and/or options specification to an installer set generator 109."); and  
23 column 7, lines 22-23 ("Step 3: The user selects desired software components and  
24 options from a list of components and options.").

1 In responding to Applicant's previous amendment and argument, the Office  
2 states, citing for support to *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194  
3 (CCPA 1958), that "[a]lthough neither Halpern nor Taylor teach that the acts of  
4 identifying and creating are performed without user input, the act of automating an  
5 otherwise manual process is obvious and does not distinguish over the prior art."

6 The logic of this argument is flawed for at least this following reason. This  
7 argument completely disregards the context of both Applicant's claimed subject  
8 matter and the cited reference, and establishes what amounts to a *per se* rule that  
9 any automation is not patentable. Applicant is unaware of any sections of the  
10 Patent Statute that establish a rule that "automation", as referred to by the Office,  
11 constitutes unpatentable subject matter. It is interesting to note that the Board of  
12 Patent Appeals and Interferences has negatively treated *In re Venner*. In addition,  
13 the Federal Circuit has expressed *per se* rules of obviousness as legally incorrect.

14 Specifically, in *Ex Parte Richard Brouillet, Jr.*, 2001 WL 1339914 (2001),  
15 the Board distinguished *In re Venner* because in *In re Venner*, all of the limitations  
16 in the claims, including the automatic means, were disclosed in the applied  
17 references. The prior art in *Ex Parte Richard Brouillet, Jr.* was missing a claim  
18 element. The Board then quotes the Federal Circuit case of *In re Ochiai*, 71 F. 3d  
19 1565, 1572, 37 USPQ2d 1127, 1133 (1995) in stating that "reliance on *per se* rules  
20 of obviousness is legally incorrect and must cease."

21 The Office admits that neither Halpern nor Taylor teach that the acts of  
22 identifying and creating are performed without user input. It is inappropriate for  
23 the Office to reject as obvious the subject matter of this claim using nothing more  
24 than a *per se* rule of obviousness, when such *per se* rules have been indicated by  
25 the Federal Circuit to be legally incorrect.

1 Accordingly, for at least this reason, the Office has failed to establish a  
2 *prima facie* case of obviousness and this claim is allowable.

3 **Claim 70** depends from claim 69 and is allowable as depending from an  
4 allowable base claim. This claim is also allowable for its own recited features  
5 which, in combination with those recited in claim 69, are neither disclosed nor  
6 suggested in the references of record, either singly or in combination with one  
7 another. In addition, given the allowability of claim 69, the rejection of this claim  
8 over the combination with Rowley is not seen to add anything of significance.

9  
10 **Claims Rejected Over the Combination of at least Bailey and Kolawa**

11 **Claim 86** recites a method of providing software extensions via the Internet  
12 comprising:

- 13
- 14 • assigning one or more files to one or more scenarios to provide  
multiple different scenarios that describe ways that a user interacts  
15 with a software application program;
  - 16 • assigning a priority to each of the scenarios;
  - 17 • sorting multiple files in accordance with their scenario priority or  
priorities; and
  - 18 • downloading sorted files in an order defined by said sorting.

19 In making out the rejection of this claim, the Office argues that Bailey  
20 teaches running a number of test scenarios on a file which represents a program  
21 (citing to column 1, lines 22-28 and lines 46-50), and assigning a priority to tests  
22 by ordering them based on coverage. The Office then goes on to argue that Bailey  
23 does not explicitly teach that the scenarios describe ways in which a user interacts  
24 with an application. The Office then relies on Kolawa and argues that it teaches a  
25 test suite which is a collection of tests that test the complete functionality of a

1 software program. Based on this, the Office surmises that Kolawa *must* describe  
2 ways that a user can use an application.

3 Continuing, the Office admits that neither Bailey nor Kolawa teach sorting  
4 multiple files in accordance with their scenario priority or downloading sorted  
5 files in an order defined by the sorting. The Office then relies on Collins and  
6 argues that it teaches such subject matter in column 5, lines 35-37. Based on the  
7 teachings of these three references, the Office argues that the subject matter of this  
8 claim is obvious. Applicant respectfully disagrees and submits that the Office has  
9 not established a *prima facie* case of obviousness for a number of different  
10 reasons.

11 First, Bailey discloses a *software testing system* that measures execution of  
12 machine code instructions in an executing program. The first two sections of  
13 Bailey cited by the Office simply describe the importance of comprehensively  
14 testing software and a particular type of tool that has been used to test software  
15 respectively. The third section of Bailey that is cited by the Office describes part  
16 of its technique for measuring the execution of machine code instructions in a  
17 computer program. It appears, from even a cursory reading of Bailey, that Bailey  
18 is not remotely associated with or concerned with methods of providing software  
19 extensions via the Internet. Thus, to this extent, Applicant continues to maintain  
20 that Bailey is not even germane to the subject matter of this claim.

21 The reference to Kolawa is no better. Kolawa is also directed to software  
22 testing and discloses a system and method that generates a test suite for a  
23 computer program that comprises program statements and variables including at  
24 least one input statement having one or more input variables that are grouped into  
25 code blocks and stored in a program database. The program statements

1 corresponding to a candidate code block are read from the program database and  
2 each input variable is represented in symbolic form as a symbolic memory value.  
3 The processing that Kolawa further describes makes it abundantly clear that  
4 Kolawa is neither directed to nor in any way concerned with methods of providing  
5 software extensions via the Internet.

6 In responding to Applicant's previous arguments, the Office states that  
7 Bailey and Kolawa are introduced to teach assigning files to scenarios that  
8 describe ways a user interacts with a software application program and assigning  
9 priorities to the scenarios. The Office then states that "Collins teaches placing  
10 software packages in a queue for downloading, thus sorting the packages in an  
11 order for downloading, where this order is taught in Bailey as a certain scenario  
12 priority order."

13 Applicant respectfully submits that the Office has engaged in impermissible  
14 hindsight reconstruction which has been specifically proscribed by the Federal  
15 Circuit. Based on the substantial differences between the subject matter of the  
16 present claim and the references to Bailey and Kolawa, the Office has failed to  
17 establish a *prima facie* case of obviousness. Accordingly, for at least this reason,  
18 this claim is allowable. In addition, based on the failure of the Office to establish  
19 a *prima facie* case of obviousness, the Office's reliance on Collins is not seen to  
20 add anything of significance. Accordingly, this claim is allowable.

21 **Claims 87-90** depend from claim 86 and are allowable as depending from  
22 an allowable base claim. These claims are also allowable for their own recited  
23 features which, in combination with those recited in claim 86, are neither disclosed  
24 nor suggested in the references of record, either singly or in combination with one  
25 another.



1  
2 **Claims Rejected Over the Combination of at least Rowley and Collins**

3 **Claim 91** recites a method of ordering files for download to a client  
4 comprising:

- 5
- 6 • sorting multiple files by one or more file groups;
  - 7 • sorting the multiple files based on scenario priority of one or more  
8 scenarios into which each file can be placed;
  - 9 • sorting the multiple files by file usage order within one or more  
10 scenarios.

11 In making out the rejection of this claim, the Office argues that Rowley  
12 teaches sorting multiple files into multiple directories, but not sorting multiple  
13 files based on scenario priority. The Office then relies on Collins and argues that  
14 it teaches sorting data packages on a queue for downloading and hence prioritizing  
15 certain packages. The Office then admits that neither Rowley nor Collins teach  
16 sorting multiple files based on file usage order. The Office then relies on Bailey  
17 and argues that it teaches ordering scenarios by priority, citing to column 8, lines  
18 21-27. Based on the teachings of these three references, the Office argues that the  
19 subject matter of this claim is obvious. Applicant respectfully disagrees and  
20 submits that the Office has not established a *prima facie* case of obviousness.

21 Specifically, Collins simply discloses that once a software package is  
22 scheduled for transmission via the internetwork to a target computer, group or  
23 Profile, an indication is stored in the Outbound Package Queue (13). See, column  
24 5, lines 34-37. Applicant respectfully submits that this in no way describes or  
25 suggests “sorting the multiple files based on *scenario priority* of one or more  
*scenarios into which each file can be placed*.”

1 As but one example of subject matter that is covered by this claim, the  
2 Office is respectfully referred to the Specification, page 31, line 1 through page  
3 33, line 13. As should be readily apparent after this example is considered,  
4 Collins in no way discloses or suggests “sorting the multiple files based on  
5 *scenario priority* of one or more *scenarios into which each file can be placed.*”

6 The Office seems to agree with Applicant on this point and argues that  
7 Bailey teaches sorting scenarios by priority. See, e.g. Office Action page 20, last  
8 paragraph. Applicant respectfully submits that this claim recites no such subject  
9 matter. That is, this claim recites:

- 10 • *sorting multiple files* by one or more file groups;
- 11 • *sorting the multiple files* based on scenario priority of one or more  
12 scenarios into which each file can be placed;
- 13 • *sorting the multiple files* by file usage order within one or more  
14 scenarios.

15 Accordingly, for at least this reason, the Office has failed to establish a  
16 *prima facie* case of obviousness and this claim is allowable.

17 **Claims 92-96** depend from claim 91 and are allowable as depending from  
18 an allowable base claim. These claims are also allowable for their own recited  
19 features which, in combination with those recited in claim 91, are neither disclosed  
20 nor suggested in the references of record, either singly or in combination with one  
21 another.

## 22 **Conclusion**

23 Applicant has made a sincere attempt to place this application in condition  
24 for allowance and respectfully requests a Notice of Allowability be issued  
25

1 forthwith. If the next anticipated action is to be anything other than issuance of a  
2 Notice of Allowability, Applicant respectfully requests a telephone call for the  
3 purpose of discussing an Appeal.  
4

5 Respectfully Submitted,

6  
7 Dated: \_\_\_\_\_

8 *6/28/04*

9 By: \_\_\_\_\_

10 *[Signature]*  
11 Lance R. Sadler  
12 Reg. No. 38,605  
13 (509) 324-9256  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25